

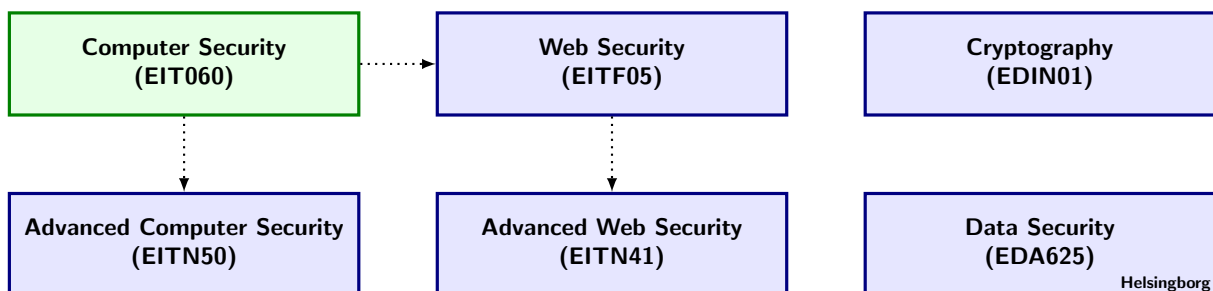
Computer Security 2015

Lab 1: Login procedure, passwords, ACLs, and security logs in Windows 7

- This lab will be done in groups of 2 people.
- There are preparatory assignments for this lab, read through the complete lab guide carefully, and bring your written answers to the lab.
- During the lab, write down answers to all problems on a sheet of paper so your work can be approved.

Learning goals:

- Get to know the Windows login procedure.
 - Know how passwords are stored in Windows, and how they can be cracked.
 - Understand the different access control settings in Windows.
 - Enable auditing to monitor security events.
-



Read this earlier than one day before the lab!

Note that you will not have any internet access during the lab, so come prepared. You may bring as many books and printed materials as you can carry. Study the questions in this lab manual, consider what you will need to be able to solve them, and make sure you bring that information with you. Alternatively, if you feel confident in the availability of eduroam, you may bring your own laptop, smartphone, or tablet to get Internet access.

There are preparatory assignments for this lab, **write down your answers, you will have to show your answers to be allowed to do the laboratory.** For most students, these assignments take more than a couple of minutes. Read through this lab guide carefully (Yes, the complete paper), and then prepare your assignments. During the lab, answer **all problems** on a separate sheet of paper, so your work can be approved.

Introduction and goal

In this laboratory lesson you will get acquainted with some different security features in the operating system Windows 7. It will cover four such features: the login procedure, passwords, file access, and the audit option. The laboratory lesson requires some preparation. You should read Chapter 4, 5 and 8 in Gollmann, Computer Security, or something similar. If you are not acquainted with Windows operating system, you should make sure you are before the laboratory lesson. Some of the preparatory assignments will require you to look for information outside the book. Any well-known search engine will turn out to be valuable then.

1 The logon procedure and security of passwords

Although recently many new technologies for authentication have been developed – such as smart cards and use of biometric data – the most commonly used method is still to authenticate a user via a username and a password. In this section we will study some authentication procedures used in Windows based networks, and see why it is important to be aware of the available weaknesses as a system administrator.

Problem 1 *What is the first sequence to type before logging on? What is so special about this sequence?*

Problem 2 *Should you type it if the logon banner is already displayed? Why?*

This computer is part of a Domain (**COMPSEC**). A domain is a group of computers, all administered by a domain controller. From the domain controller, the administrator can manage each computer and every user/group. When logging in, Windows 7 will by default ask for the password of the last logged in user. If you want to log in as another user, you must first click *Switch user* followed by *Other User*.

By default, the computer will try to authenticate against the domain, which can be seen on the logon screen as “Log on to: COMPSEC”. You can also login to the local domain by prepending **ACE-##** in front of the username, where **ACE-##** is the computer name as indicated on the computer. However, in this lab you will use domain accounts. Login to COMPSEC using:

User name: lina##
Password: Kanejbytas123

This makes the login manager in Windows contact the domain controller for an authentication of your account.

Check who belongs to the local group **Administrators** by right-clicking on **Computer** in the Start Menu, and select **Manage**→**Local users and groups**→**Groups** (your account is a part of the group "**CompsecUsers**"). Check which **Users** have access to the computer. As you can see there are several local accounts (not domain accounts) on your computer called **labuserXY**, where X is a number in $\{1, 2\}$ and Y is a letter in $\{a, \dots, f\}$. There are also six **prouserZ** accounts. For example, labuser1g, labuser2a, and prouser3 are three accounts. A local user or group is an account that can be granted permissions and rights from your computer. Domain or global users and groups are managed by your network administrator.

1.1 Browser passwords: Mozilla Firefox

The major browsers today have a feature to save the user's passwords on websites, so that they do not have to be entered every time. In this lab, we will look at how the passwords in two popular browsers—Microsoft Internet Explorer and Mozilla Firefox—can be accessed.

We will first look at Mozilla Firefox: Start Firefox by clicking **Start**→**All programs**→**Mozilla Firefox**. Go to the options menu, by clicking **Orange Firefox-button**→**Options**. Now, click on the tab *Security* and the button *Saved Passwords*.

■ Problem 3 *What passwords can you find?*

Even though you do not have to verify it in this lab, it is worth noting is that passwords in Google Chrome are saved in a similar fashion. There you can find the passwords by going to the Preferences menu, scroll down and click *Manage saved passwords*, and then view any password by clicking on it and select *Show*.

However, if we try to find a similar settings page in Microsoft Internet Explorer, we will not find any. To be able to look at Internet Explorer's passwords, we need to use an external utility. There are multiple tools available, but in the next section we will look at one general purpose tool which can be used for several password related tasks in Windows.

1.2 Cain & Abel

Cain & Abel is a password recovery tool for Windows. It allows easy recovery of various kind of passwords by sniffing the network; cracking encrypted passwords using Dictionary, Brute-Force, and Cryptanalysis attacks; recording VoIP conversations; decoding scrambled passwords; recovering wireless network keys; uncovering cached passwords; and analyzing routing protocols.

Preparatory assignment 1

- Read about Cain & Abel such that you are comfortable working with the tool.
(see http://www.oxid.it/ca_um/ for user manual, currently only works in Firefox)
- What is a brute force attack?
- What is a dictionary attack?
- What is a time memory tradeoff attack?
(see e.g., <http://www.cs.miami.edu/~burt/learning/Csc609.122/doc/36.pdf>)
- What is the difference between a rainbow table and an ordinary time memory tradeoff table? (see e.g., <http://lasecwww.epfl.ch/pub/lasec/doc/0ech03.pdf>)

In this laboratory we will focus on the Protected Storage recovery and the Hash Cracking utilities in Cain & Abel.

Start Cain & Abel, i.e., **Start**→**All Programs**→**Cain**→**Cain**. A warning from the Windows User Account Control will show, since Cain & Abel requires Administrator rights. Allow this, and ignore the next warning about Windows Firewall.

1.2.1 Browser passwords: Internet Explorer

We will start by studying the tab named **IE 7/8/9**. This tab can be used to view the stored passwords of Internet Explorer, which are actually stored in the Windows registry.

Highlight **IE 7/8/9**. Click the blue + icon to dump the passwords into the tab.

| Problem 4 *What passwords do you find?*

There are many other decoders available in Cain, such as: *Protected storage* which stores saved passwords for older version of Internet Explorer, and *Credential Manager* which for example stores Microsoft Outlook passwords.

You have now seen how different browsers handle the stored passwords.

| Problem 5 *Does any browser store the passwords more securely than the other? Explain!*

As you can see, the passwords in all three browsers are readable in clear-text, this means that leaving your computer unattended for only a short time may compromise your passwords if an attacker is nearby!

1.2.2 Cracking LM/NTLM Hash Values

Cain includes a cracker for different kinds of hash functions. Basically, three kind of attacks are supported: brute force, dictionary, and rainbow table attacks (called *Cryptanalysis attack* in Cain).

Preparatory assignment 2

- How is the LAN Manager (LM) hash produced, how is this hash used to authenticate a user, and what is the effective security of LM?
- How is the NT LAN Manager version 1 (NTLM) hash produced, how is this hash used to authenticate a user, and what is the effective security of NTLM?
- How does a challenge-response protocol work?

User records are stored in the security accounts manager (SAM) database for local accounts, and in the Active Directory database for domain users. Passwords are hashed and also stored in the SAM database or in the Active Directory database together with the user record.

By default, Windows XP and previous versions of Windows stored both the LM and NTLM hash for backward compatibility reasons, i.e., both hashes was stored in the SAM file and Active Directory. This was a devastating mistake as we will soon see, and as an administrator of older systems it is very important to know about this fact.

Starting with Windows Vista and Windows Server 2008, the calculation of the LM hash is disabled by default, but it can still be activated by changing a registry value. In this lab, some passwords will – for demonstration purposes – have a calculated LM hash even though we use Windows 7.

On your computer there are several local users called **labuserXY** whose passwords you should try to crack. The accounts **labuser1Y** will have LM-hashes in addition to NTLM, the other accounts will only have NTLM hashes. Choose at least three of the users, at least one from each *X*-number category. There are also some accounts called **prouserZ**, ignore them as of now, we will deal with them soon.

Highlight the tab named **Cracker** and the **LM&NTLM Hashes**. Click the blue + icon to dump all hash values from the local SAM file into the tab. Right click on a user to see different attack options.

To crack the passwords we are going to use both dictionaries and rainbow tables. Rainbow tables can be found in: **C:\Computer security\Rainbow Tables** and dictionaries in: **C:\Computer security\Dictionaries**.

Note: Before you start cracking using dictionaries, make sure you right click on any of the dictionaries and choose the option *Reset all initial file positions*.

| **Problem 6** Find the passwords of your chosen accounts.

| **Problem 7** Which password(s) can be found by dictionary attacks? Which password(s) can be found using the rainbow table. Explain the results.

| **Problem 8** Give a password that would not be possible to crack in this way.

Try to find the password from any of the **prouserZ** accounts using rainbow tables and dictionaries as above. Can you find any? Also try to use a *Brute-force*-attack, assuming the password is 7 characters in the range [a-zA-Z0-9].

| **Problem 9** How long would it take to try all passwords using brute-force?

1.2.3 MS-CACHEv2 Hashes

By default, Windows stores a hashed copy of domain logon passwords into the local registry; this enables the user to logon locally even if the domain controller is offline or unavailable. The passwords are hashed and then encrypted with the NL\$KM LSA secret before being stored in the registry. Cain's MS-Cache Hash Dumper allows you to import password hashes directly into the *MS-Cache Hashes* password cracker tab. The Hash Dumper feature decrypts the cached hashes and prepares them to be cracked using Dictionary, Brute-Force and Rainbow attacks.

Rainbow tables are pretty fast however only useful to crack some kind of encrypted passwords. In challenge-response authentication protocols the challenge and response are always different and hence makes rainbow tables useless.

Another protection is to use a *salt*. This is a constant value that is added to the password to make sure that two users who have the same password does not end up with the same hash. It also makes rainbow table attacks less efficient since a rainbow table has to be made for each salt. However, since the salt is a known constant, rainbow table attacks can be efficient if the salt is predictable enough.

Highlight **MS-Cache Hashes**. Click the blue + icon to dump all users who has logged onto your computer.

| **Problem 10** Launch a brute-force attack on any of the MS-CACHEv2 hashes you find. Use the same character set and password length as before. How long time would it take to test all passwords? You do not have to wait for it to finish.

An important difference between LM/NTLM hashes and MS-Cache hashes is that the MS-Cache hashes use a salt. Because of this, Rainbow attacks can not be used as easily as for LM/NTLM hashes. However, in Windows the salt value used is pretty predictable, namely it is the username. So a table can be built for the special salt value of the victim. The table can then only be used for accounts with that name.

| Problem 11 *Is the salt well chosen in Windows? Why or why not?*

Preparatory assignment 3

- Ensure that you understand what a salt is, and why it is used.
- Find information and read about PBKDF2. Roughly compare the time required to calculate a hash with PBKDF2 with the time required for a single round of SHA-1. You do not have to understand the algorithm in detail.

Starting with Windows Vista, the algorithm for cached domain credentials has been modified to use PBKDF2. We will look at the properties of this algorithm later in this lab.

1.3 Cracking passwords using GPUs

In the previous examples the CPU of your local computer has been used to try different password combinations. In this assignment, you will instead use a separate lab computer (hereafter called **cracker**) which is equipped with several graphics processing units (GPUs). You will use these GPUs to crack passwords.

Hashcat is another well-known application used when dealing with password hashes. There are two different versions available: hashcat and oclHashcat. The different versions support mostly the same hashes, but oclHashcat uses GPUs, while hashcat only supports CPUs. During the lab **cracker** will run Linux, will be equipped with *two* AMD HD7990 graphics cards, and use oclHashcat to crack the passwords.

Preparatory assignment 4

- Read about hashcat and oclHashcat such that you have a rough idea of the features. Pay attention to where you can find performance figures for CPUs and GPUs using different hashes (<https://hashcat.net/oclhashcat/>, <https://hashcat.net/hashcat/>).
- How many password combinations can be tried every second on a CPU using hashcat? Assume that NTLM hashes are used, and that the processor is AMD FX(tm)-8120 Eight-Core Processor using 8 threads.
- Estimate the number of password combinations that can be tried every second using the GPUs on **cracker**. You may assume that a *single* AMD HD7990 is equivalent to two AMD HD7970, and that NTLM hashes are used.
- Assume you want to brute force a password of length 8, with characters in the range [a-zA-Z0-9]. Estimate the time required to test all passwords with the CPU above. How about the time required using **cracker**?

You will now try to break some hashes using the **cracker** computer. We will try the same brute-force attack as we tried in Cain. *Read through the problems described below once again*, such that you know what to look for when running oclHashcat. Go to the computer with the GPUs and launch the scripts below, which are located in `~/labscripts/`.

Problem 12 Run the script `./crack-ntlm.sh`. You may press the key `s` to print the current status. How many combinations are tried every second? Was your estimate from the preparatory assignment correct?

Problem 13 Which passwords do you find? Cracked passwords will be printed on screen as `hash:password`. Note that you may have to scroll if you pressed `s` in the problem above.

Problem 14 Now try to run the script `./bench-mscash2.sh`. This will run a short benchmark with the algorithm used in domain cached credentials described in the previous section. How many combinations are tried every second? Any difference compared to the NTLM algorithm?

When you are finished at the **cracker** computer, you may enter the command `reset` in the terminal to erase the history, and prevent the next student from looking at your cracked hashes. Please do not close the terminal windows, but if you do it by accident, you can open a new one by pressing *Win-Enter*.

You are now finished with the first part of the laboratory. Check your answers before you continue with the next part.

2 Windows access control

Next, we are going to play with the most important security features of Windows 7—its access control lists (ACLs). To do so you must be familiar with creating and managing folders and files in Windows 7. The best way to handle this is to use Windows Explorer. So let's get started. Create a folder in the directory `C:` that has the same name as your account `lina##` (e.g. `lina48`). Create a text file `test` in this directory. Have a look at the default permissions determined by the ACL for the directory `lina##`. This is done by clicking on `lina##` and then selecting **Properties**→**Security**.

Problem 15 Who owns the directory `lina##`? Who can (according to the ACL) access the directory and what can they do?

Currently, your directory `lina##` is not available on the local network. To make it available for other users on other computers you need to share it to the network. Still in the Properties windows, go to the tab **Sharing**. Click **Advanced sharing...**, check the box next to **Share this folder** and look at the share permissions for your directory.

Problem 16 Who can (according to the share permissions) access the directory and what can they do?

Set the share permissions to grant *Everybody* full control, and click OK to close the share dialog. Now your directory is shared on the network. You can check which directory you share to the network by looking in the Explorer under Network. The ACE-# computers are sometimes connected through a firewall to the rest of the network. Then you might have to write the path to the computer you are looking for, i.e., write `\\ACE-#` in the address bar.

Now it is time to start messing around with your ACLs. When you create a new directory, the default permissions for the directory is inherited from the directory in which you create the new directory. This can sometimes be confusing so let us clean up the ACL a bit. For your `C:\lina##` directory, choose the **Security**→**Advanced**→**Change permissions** and uncheck the box for *Include inheritable permissions from this object's parent*. Choose *Remove* in the next dialog, and then add your account with *Full control*. Next add the group *Everyone* and mark it such that it is denied all access. Click Yes when the warning appears.

| **Problem 17** *Can you access the file? What rule does Windows 7 apply here?*

Now remove *Everyone* from the ACL of the folder `lina###`. Note that if the *Replace all child object permissions with inheritable permissions from this object* is checked, then the permissions changed on the directory will also be set to all the files in the folder, including the text file. Instead of *Everyone*, you should add the group *CompsecUsers*, which is a group account where all `lina##` accounts are members.

| **Problem 18** *Can your neighbor read the file? Give your neighbor Full control to the file. Can your neighbours read the file? Can they write to the file? Can they change the permissions on the file? Can they take ownership of the file?*

As you see giving someone *Full control* is a very dangerous thing. Instead you can use the special access, this lets you change the permissions anyway you like. You access the special permissions from the advanced button on the security tab.

| **Problem 19** *Try some different rules and see that it works.*

| **Problem 20** *What ACLs do newly created files in the directory get? Try some different settings on the directory permission and create some files in the directory.*

Now set the sharing permission on the `lina##` folder to *read* for everyone. Then set the security permission to the file and the folder to *Full control* for everyone.

| **Problem 21** *Can you write to the file and can your neighbour write to the file? Experiment with different permissions for the folder/file and the sharing permissions. What rules are used here?*

Hopefully, you are now quite familiar with ACLs and related topics.

You are now finished with the second part of the laboratory. Check your answers before you continue with the next part.

3 Security Logs

The next topic we introduce is the security auditing features of Windows 7. As your account is a local administrator, you are allowed to manage security logs on your computer.

The first tool you need to use is the **Event viewer** which is found in the Administrative Tools folder by clicking **Start→Control Panel→Administrative Tools→Event Viewer**. Clear the Event viewer by right-clicking **Windows Logs→Security** and selecting **Clear Log**. Don't save any events to disk.

To enable auditing, open **Local Security Policy**, which is also found in the **Administrative Tools** folder. Set the audit policy to include both successful and failed logons by selecting **Local Policies→Audit Policies→Audit logon events**.

Log off and try to logon with incorrect password. Then logon correctly.

Look at the Event viewer. Double-click on individual events for details.

| **Problem 22** *Why does the security log not show the incorrect password that you used during the failed logon attempt?*

In the final exercise you will audit file and object access. Add success and failure logging of **Audit object access** in the **Audit Policy** dialog. Note that you will have to place audit control on the

individual file or folder that you want to audit. By a right mouse click you select **Properties**→**Security**→**Advanced**→**Auditing** and then select the events that you want to audit. Let your neighbor perform an access to your file/folder and check the Event viewer.

If you generate too much audit information, check out how to use the **View**→**Filter Options** in the Event Viewer.

4 Cleanup

Restore the computer for the next laboratory lesson.

- ☐ Clear all events in the Event Viewer.
- ☐ Turn the auditing off.
- ☐ Remove your folder `lina##` and all its content.
- ☐ In Cain, remove your cracked `labuserXY` accounts. You can do this by right-clicking in the account list and choose *Remove All*.

You are now finished, check your answers before contacting your friendly teaching assistant.

A Appendix: Meaning of permissions in NTFS

Permission	Description
Traverse Folder/Execute File	For folders: Traverse Folder allows or denies moving through folders to reach other files or folders, even if the user has no permissions for the traversed folders (applies to folders only). Traverse folder takes effect only when the group or user is not granted the Bypass traverse checking user right in the Group Policy snap-in. (By default, the Everyone group is given the Bypass traverse checking user right.) For files: Execute File allows or denies running program files (applies to files only). Setting the Traverse Folder permission on a folder does not automatically set the Execute File permission on all files within that folder.
List Folder/Read Data	List Folder allows or denies viewing file names and subfolder names within the folder. List Folder only affects the contents of that folder and does not affect whether the folder you are setting the permission on will be listed. Applies to folders only. Read Data allows or denies viewing data in files (applies to files only).
Read Attributes	Allows or denies viewing the attributes of a file or folder, such as read-only and hidden. Attributes are defined by NTFS.
Read Extended Attributes	Allows or denies viewing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.
Create Files/Write Data	Create Files allows or denies creating files within the folder (applies to folders only). Write Data allows or denies making changes to the file and overwriting existing content (applies to files only).
Create Folders/Append Data	Create Folders allows or denies creating folders within the folder (applies to folders only). Append Data allows or denies making changes to the end of the file but not changing, deleting, or overwriting existing data (applies to files only).
Write Attributes	Allows or denies changing the attributes of a file or folder, such as read-only or hidden. Attributes are defined by NTFS. The Write Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the attributes of a file or folder. In order to allow (or deny) create or delete operations, see Create Files/Write Data, Create Folders/Append Data, Delete Subfolders and Files, and Delete.
Write Extended Attributes	Allows or denies changing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program. The Write Extended Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the attributes of a file or folder. In order to allow (or deny) create or delete operations, see Create Files/Write Data, Create Folders/Append Data, Delete Subfolders and Files, and Delete.

Permission	Description
Delete Subfolders and Files	Allows or denies deleting subfolders and files, even if the Delete permission has not been granted on the subfolder or file. (applies to folders)
Delete	Allows or denies deleting the file or folder. If you don't have Delete permission on a file or folder, you can still delete it if you have been granted Delete Subfolders and Files on the parent folder.
Read Permissions	Allows or denies reading permissions of the file or folder, such as Full Control, Read, and Write.
Change Permissions	Allows or denies changing permissions of the file or folder, such as Full Control, Read, and Write.
Take Ownership	Allows or denies taking ownership of the file or folder. The owner of a file or folder can always change permissions on it, regardless of any existing permissions that protect the file or folder.